بسم الله الرحمن الرحيم

Fuzzy Control Course

Lec 9

Differential Evolution (DE) Optimization Algorithm

DR. M. Arafa

13/12/2016

The Basics of Differential Evolution

- population-based optimization algorithm
- Introduced by Storn and Price in 1997.
- Developed to optimize *real parameter*, *real valued* functions.
- General problem formulation is:

For an objective function $f : X \subseteq R^D \rightarrow R$

where the feasible region $X \neq \emptyset$, the minimization problem is to find

$x^* \in X$ such that $f(x^*) \leq f(x) \; \forall \, x \in X$

Where:

$$f(x^*) \neq -\infty$$

The Basics of Differential Evolution

- DE is a parallel direct search method which utilizes NP D-dimensional parameter vectors.
- Suppose we want to optimize a function with D dimensional real parameters R^D.
- We must select the size of the population NP (NP must be ≥ 4),

$$x_{i,G} = [\, x_{1,i,G}, \, x_{2,i,G}, \, . \, . \, . \, x_{D,i,G} \,]$$

$$i = 1, 2, . . . , NP \text{ and } j = 1, 2, . . . , D$$

where $x_{i,G}$ is a parameter vector in a population for each generation and G is the generation number.

- NP does not change during the minimization process.

Differential Evolution (DE)

- DE is an Evolutionary Algorithm.
- DE is constructed from initialization and a cycle of stages of mutation, recombination (or crossover), and selection.

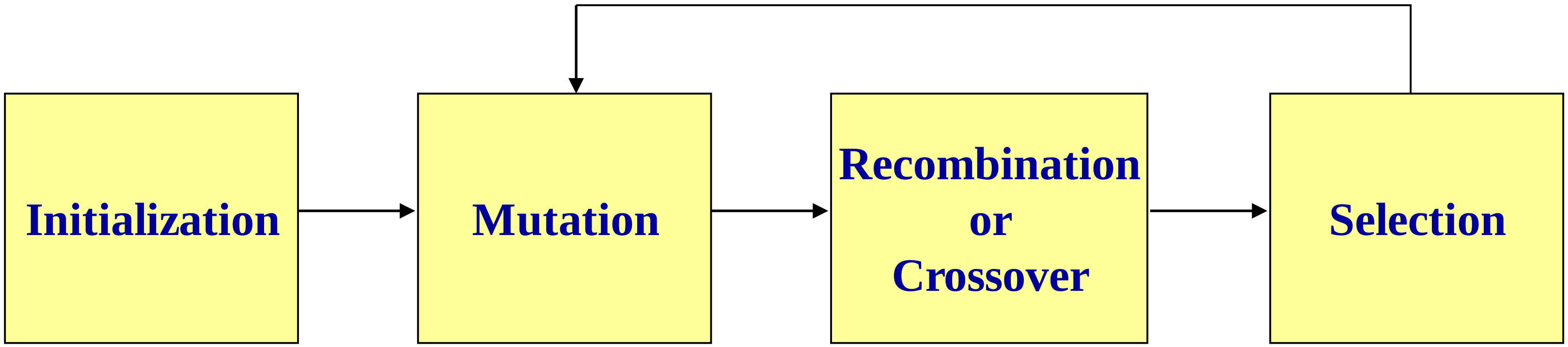
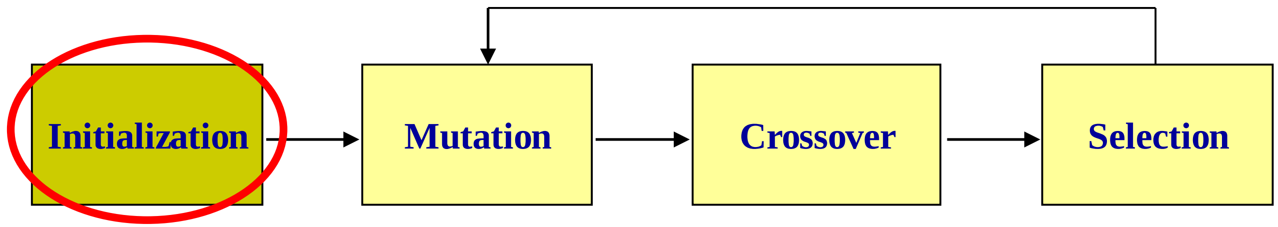


Figure 1: Basic Stages of DE

Initialization



- all parameter vectors in a population are *randomly initialized*.
- Define upper and lower bounds for each parameter:

$$x_j^L \leq x_{j,i,1} \leq x_j^U$$

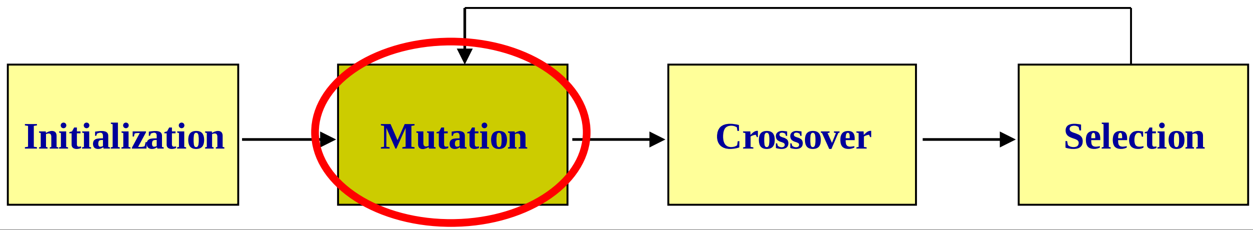
- Randomly select the initial parameter values uniformly on the intervals

$$[\ x_j^L\ ,\ x_j^U]$$

- Suggestion to chose random values between high bound and low bound

$$x_{j,i,1} = x_j^L + \text{rand}*(x_j^U - x_j^L)$$

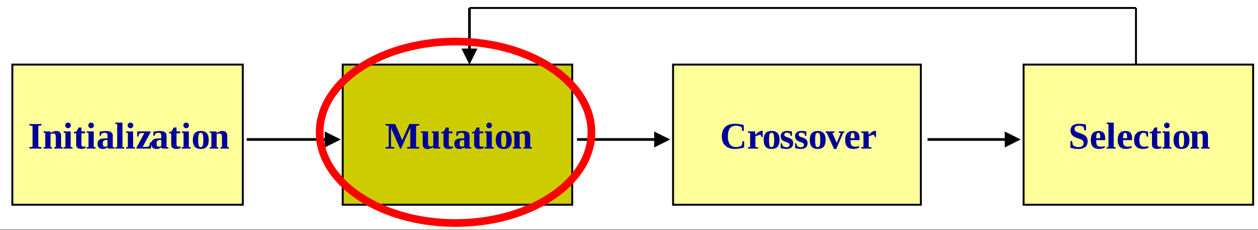
Mutation



- Mutation, recombination and selection will be run for each of the *NP* parameter vectors of a population.
- For a given parameter vector $x_{i,G}$ (called target vector) randomly select three vectors $x_{r1,G}$, $x_{r2,G}$ and $x_{r3,G}$ such that the indexes *i, r1, r2* and *r3* are distinct integers $\in \{1, 2, \ldots, NP\}$.
- Add the weighted difference of the two vectors $x_{r2,G}$, $x_{r3,G}$ to the base vector $x_{r1,G}$.

$$v_{i,G+1} = x_{r1,G} + F.(x_{r2,G} - x_{r3,G})$$

Mutation



- The mutation factor F is a constant from [0 , 2] which controls the amplification of the differential variation $(x_{r2,G} - x_{r3,G})$.
- $v_{i,G+1}$ is called the *mutant* vector or *donor* vector.

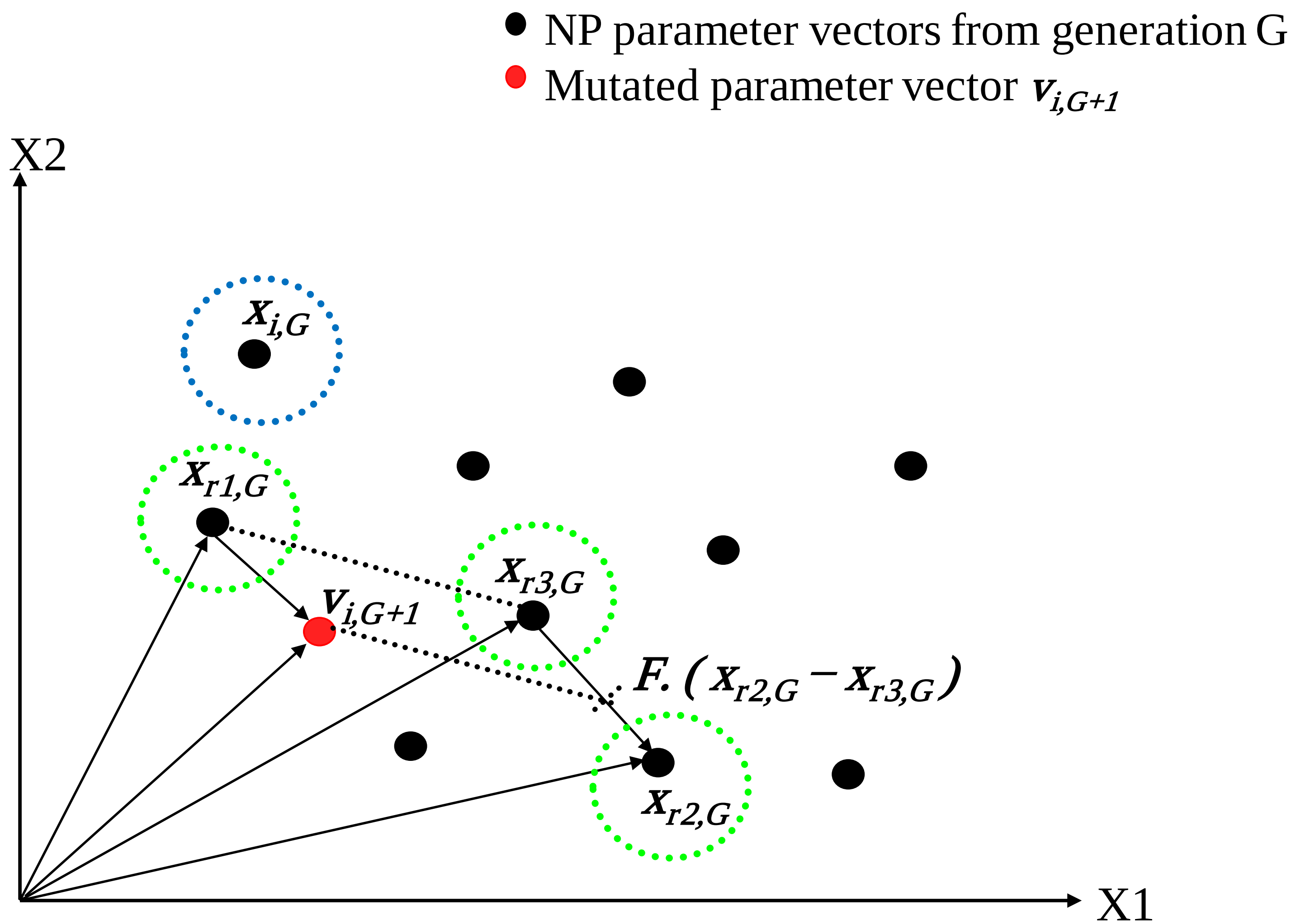
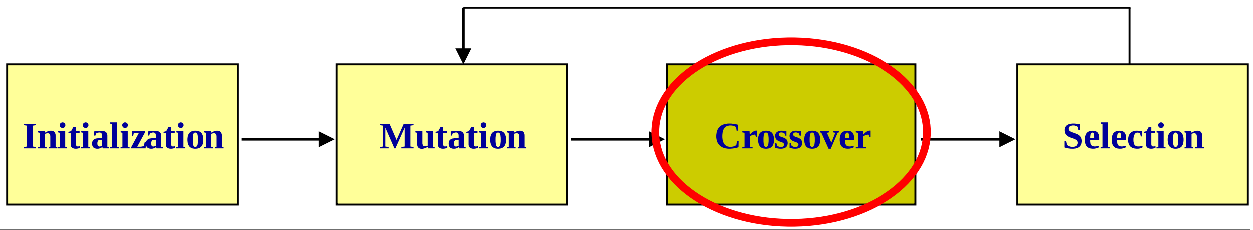


Figure 2: An example of a two-dimensional cost function showing the process for generating the mutant vector $v_{i,G+1}$ by three different vectors $x_{r1,G}$, $x_{r2,G}$ and $x_{r3,G}$.

Crossover

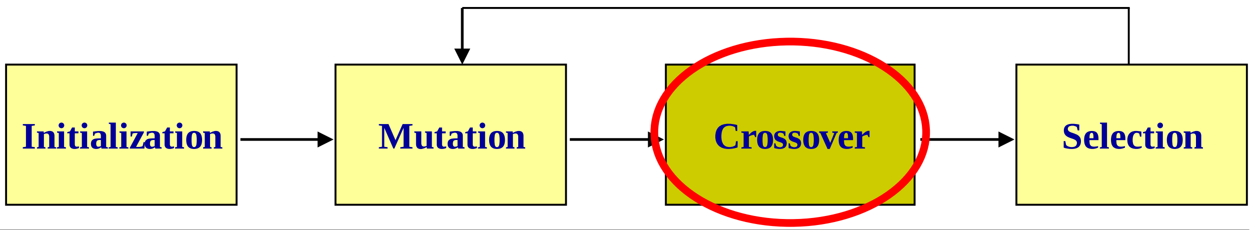


- Crossover incorporates successful solutions from the previous generation.
- The trial vector $u_{i,G+1}$ is developed from the elements of the target vector $x_{i,G}$ and the elements of the mutant vector $v_{i,G+1}$.

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } (randb(j) \leq CR) \text{ or } j = Irand \\ x_{j,i,G} & \text{if } (randb(j) > CR) \text{ and } j \neq Irand \end{cases},$$

$$i = 1,2,\ldots,NP;\ j = 1,2,\ldots,D.$$

Crossover



- *randb(j)* is the j^{th} evaluation of a uniform random number generator with out come $\in [0 , 1]$.

- *CR* is called "the crossover rate" and it is a constant $\in [0 , 1]$ which has to be determined by the user.

- *Irand* is a random integer from [1, 2, ...,D] which ensures that the trial vector $u_{i,G+1}$ gets at least one parameter from $v_{i,G+1}$.

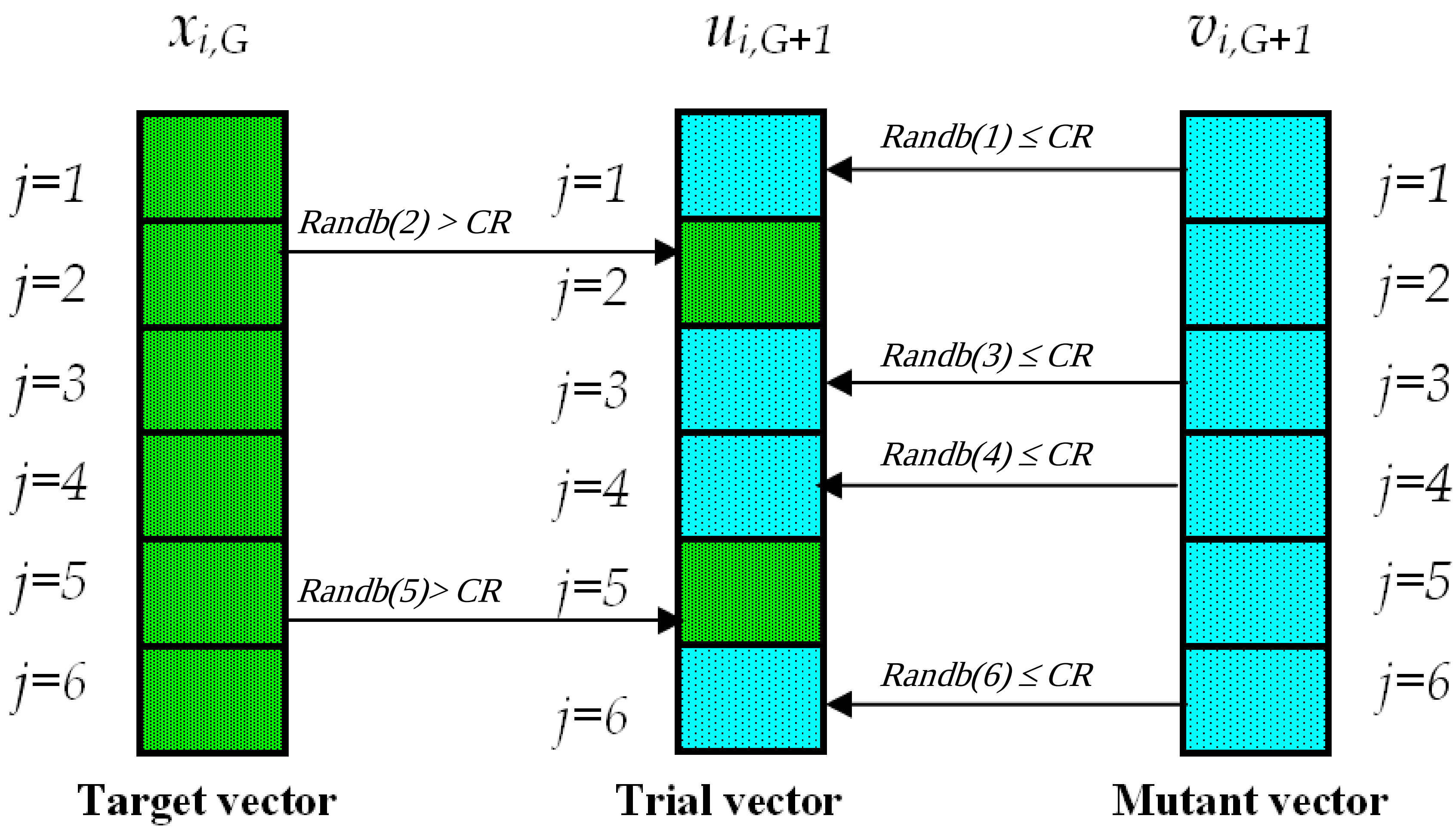
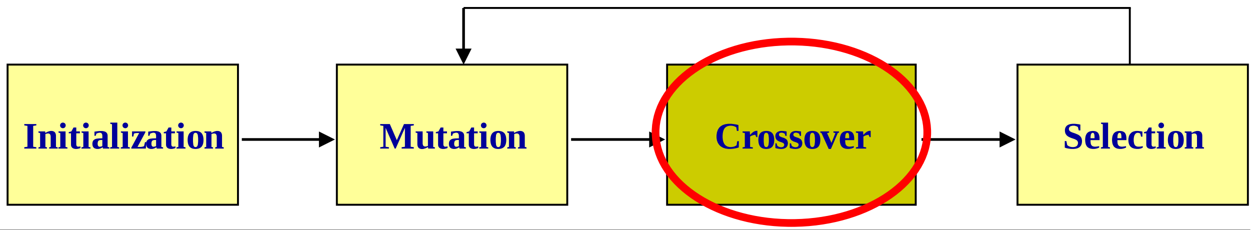


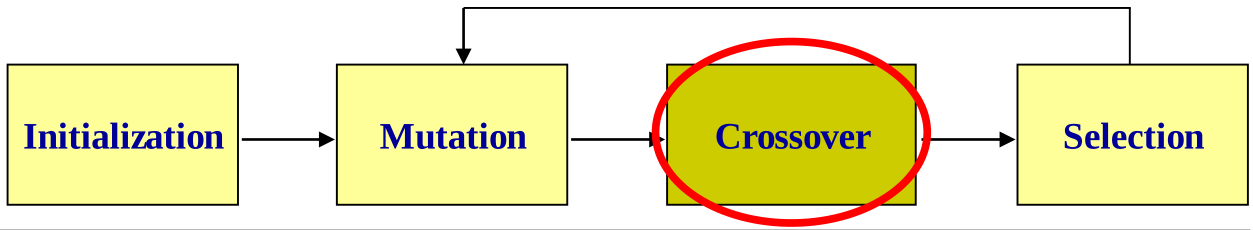
Figure 3: Illustration of the crossover process for $D = 6$ parameters.

Crossover



- After the crossover process, some or all the components of the trial vectors may be lie outside the search domain.
- So, to be sure that all components of the generating trial vectors over all iterations are within the predefined boundary constraints, one of the following three equations is used:

Crossover



$$u_{j,i,G+1} = \begin{cases} x_j^U + rand_{j,i}.\,(x_{j,i,G} - x_j^U)\ , \text{ if } (u_{j,i,G+1} > x_j^U) \\ x_j^L + rand_{j,i}.\,(x_{j,i,G} - x_j^L)\ , \text{ if } (u_{j,i,G+1} < x_j^L\ ) \end{cases} \qquad (1)$$

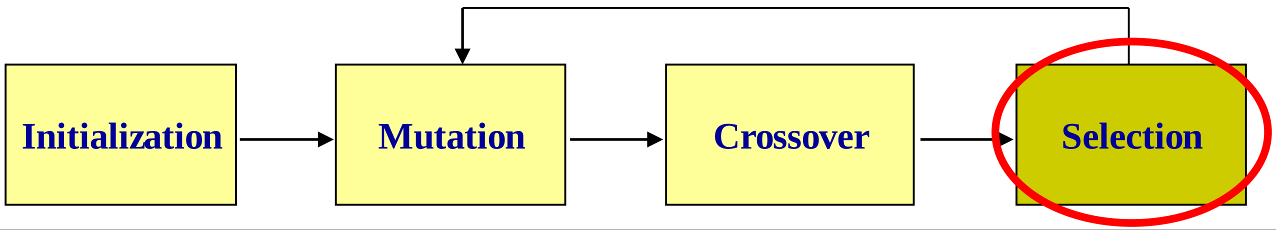
or,

$$u_{j,i,G+1} = \begin{cases} 2x_j^U - u_{j,i,G+1}\ , \text{ if } (u_{j,i,G+1} > x_j^U) \\ 2x_j^L - u_{j,i,G+1}\ , \text{ if } (u_{j,i,G+1} < x_j^L) \end{cases} \qquad (2)$$

or,

$$u_{j,i,G+1} = \begin{cases} (x_j^U + x_{j,i,G})/2\ , \text{ if } (u_{j,i,G+1} > x_j^U) \\ (x_j^L + x_{j,i,G})/2\ , \text{ if } (u_{j,i,G+1} < x_j^L) \end{cases} \qquad (3)$$

Selection



- The target vector $x_{i,G}$ is compared with the trial vector $u_{i,G+1}$ and the one with the lowest cost function value is chosen to the next generation

$$x_{i,G+1} = \begin{cases} u_{i,G+1}\text{ , if } \left(f\left(u_{i,G+1}\right) \le f\left(x_{i,G}\right)\right) \\ x_{i,G}\text{ , "otherwise"} \end{cases}$$

- Mutation, recombination and selection continue until some stopping criterion is reached

Meaning of the name?

DE/rand/1/bin

- **DE**: Differential Evolution
- **rand**: Base vector for mutation is chosen randomly
- **1**: one difference vector is used to construct the donor
- **bin**: crossover is binomial

Other Variants of DE Schemes

The mutant vector $v_{i,G+1}$ is generated according to one of the following equations:

- $v_{i,G+1} = x_{r1,G} + F.(x_{r2,G} - x_{r3,G})$ (1)

DE/rand/1/bin

- $v_{i,G+1} = x_{best,G} + F.(x_{r1,G} - x_{r2,G})$ (2)

DE/best/1/bin

- $v_{i,G+1} = x_{i,G} + F.(x_{best,G} - x_{i,G}) + F.(x_{r1,G} - x_{r2,G})$ (3)

DE/target-to-best/1/bin

Other Variants of DE Schemes

- $v_{i,G+1} = x_{best,G} + F.(x_{r1,G} - x_{r2,G}) + F.(x_{r3,G} - x_{r4,G})$ (4)

DE/best/2/bin

- $v_{i,G+1} = x_{r1,G} + F.(x_{r2,G} - x_{r3,G}) + F.(x_{r4,G} - x_{r5,G})$ (5)

DE/rand/2/bin

DE Control Parameters

- The parameters that control the performance of DE are three:

  (1) The population size ***NP***

  (2) The mutation factor ***F***

  (3) The crossover rate ***CR***

- These parameters should be chosen (or tuned) carefully to avoid the state of stagnation (or premature convergence) for the DE algorithm.

DE Control Parameters

- NP affects the ability to search the parameter space.
  - NP must be ≥ 4 (why?).
  - Small values of NP result in few numbers of mutant vectors that may cause insufficient exploration (premature convergence).
  - On the other hand, large values of NP result in many numbers of mutant vectors that may cause excessive exploration (slow convergence) and increase the number of computations.
  - In [4] $NP = 30$ for all small dimension values $D < 30$ and $NP = D$ for large dimension values $D \geq 30$.

DE Control Parameters

- The mutation factor F is relevant to the convergence speed as it is responsible for the step size that interferes in the formation of the mutant vector.
  - Small values of F will lead to premature convergence
  - $F > 1$ will try to take large steps, leading to slow convergence.
  - A good initial choice of F is 0.5 and the effective range usually lies in [0.4, 1] as Storn and Price suggested in [1].

DE Control Parameters

- The crossover rate *CR* controls the number of changes in parameters of a population member.
  - A small value of *CR* ('strong' crossover e.g. 0 or 0.1) leads to most changes occurring along one dimension or a small subset of dimensions, and this is useful for separable functions.
  - Large values of *CR* (near 1) lead to most components being chosen from the mutant vector and this is suitable for non-separable functions.
  - In [5], *CR* lies in [0, 0.2] when the function is separable and lies in [0.9, 1] when it is non-separable.

References

[1] Storn, R. and Price, K. (1997), ‘Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces’, Journal of Global Optimization, 11, pp. 341–359.

[2] Price, K., Storn, R., Lampinen, A., “Differential Evolution: A Practical Approach to Global Optimization”, Springer-Verlag Berlin Heidelberg 2005.

[3] Uday K. Chakraborty, ” Advances in Differential Evolution”, Springer-Verlag Berlin Heidelberg 2008 .

[4] Noman, N., and Iba, H., "Accelerating differential evolution using an adaptive local search", IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, pp. 107-125, Feb. 2008.

[5] Ronkkonen J., Kukkonen, S., and Price, K., "Real-parameter optimization with differential evolution", in Proc. IEEE CEC, vol. 1, pp. 506–513, 2005.